

Tecnológico de Monterrey

Campus Querétaro

**Object Oriented Programming  
Group 3**

**Integrative Project (Proposed solution to the problem situation)**

Teacher

Silvana De Gyvés Avila

Presents

Pablo César Jiménez Villeda A01703517

Delivery date: June 4th 2021

## Contents

<b>Introduction .....</b>	<b>3</b>
<b>UML Diagram: .....</b>	<b>4</b>
<b>Design justification: .....</b>	<b>4</b>
<b>Execution example .....</b>	<b>6</b>
<b>Solution justification .....</b>	<b>12</b>
• The proper classes are identified .....	12
• Inheritance is implemented properly .....	13
• Access modifiers are implemented properly .....	13
• Method overriding is implemented properly .....	14
• Polymorphism is implemented properly .....	14
• Abstract classes are implemented properly .....	15
• At least one operator is overloaded properly .....	15
<b>Functionality problems .....</b>	<b>16</b>
<b>Conclusion .....</b>	<b>23</b>
<b>References: .....</b>	<b>24</b>

## Introduction

Recently a new trend has appeared around the world, which is to distribute audiovisual content through paid streaming services. These new services are currently replacing the traditional media as they appeal to new generations and gain popularity because of the nuances they have introduced into the entertainment media.

The challenge is to propose a solution to a problem situation that involves said services. Which is that in recent years, low-cost, on demand streaming services as Netflix, Disney or DC have become popular. Some of these services are focused on the volume of videos made available to users, whereas others are challenged to show only their own branded content. A limited version to support a future content provider for these kinds of services is described next.

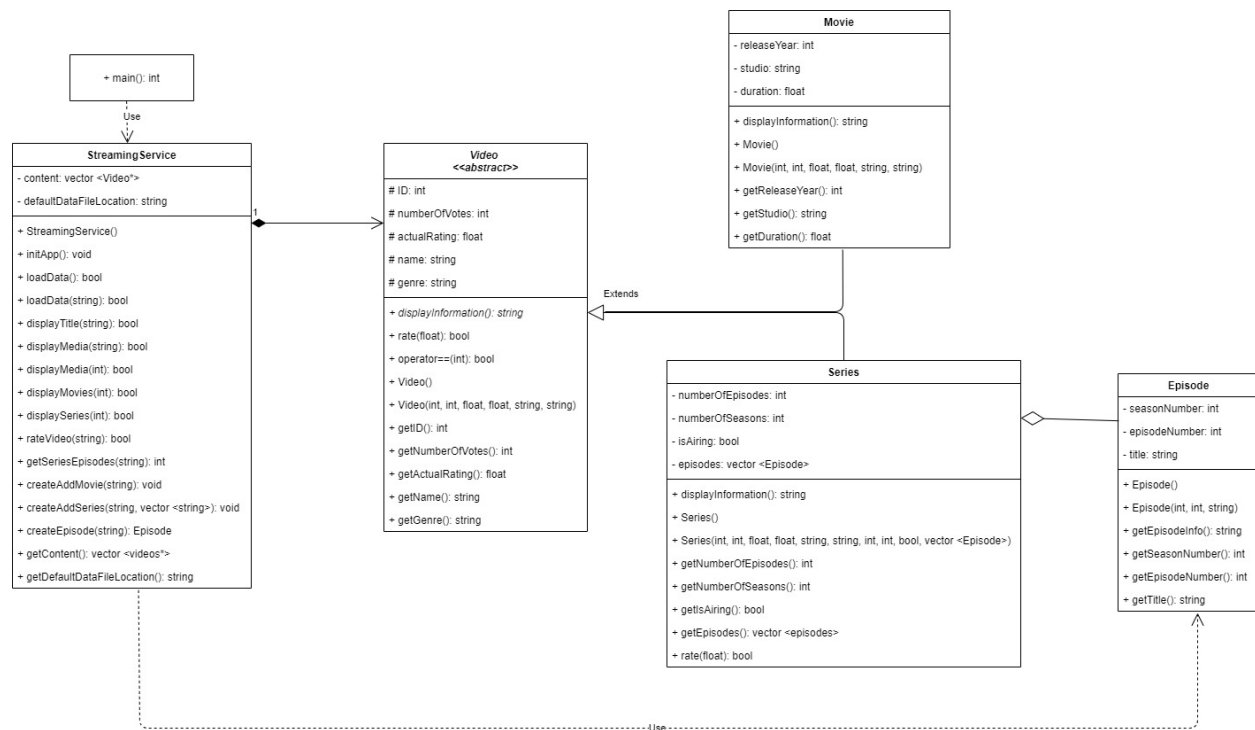
The provider will work with two kind of videos: movies and series. Every video has an ID, a name, a length and genre, i.e. drama, action, mystery. Series have episodes and each episode has a title and a season to which it belongs to. We're interested in knowing the average rating that each of the videos has received. This rating is in the scale from 1 to 5, where 5 is the best rating.

The required computational system to manage the videos will be capable of the following:

- Showing the collection of videos by title, along with their ratings.
- Showing the episodes for a given series, along with their ratings.
- Showing the movies, along with their ratings.

The proposed solution for this computational system will be detailed next as well as the UML diagram that represents the implementation of the Object Oriented Programming concepts reviewed in the course.

## UML Diagram:



## Design justification:

From the problem statement and the information provided I came to the conclusion that I would have to build a solution using several Object Oriented Programming concepts, for instance, seeing the relations between the different types of data mentioned in the introduction I decided that I had to use inheritance and implement 5 classes to correctly represent the components of a streaming service and create a console simulation of a rating and search system. In order to correctly structure a solution an UML diagram to represent the relations between the components of the program was needed for reference. When taking into account the nuances of the problem the proposed UML diagram was the one seen in the previous figure.

As it can be seen, there will be a class called StreamingService used by the main file, the reasoning behind this was that the main file would serve as a starting point for any user that needed to tinker with the data inside of a file. For this, an instance of this class would be created and then through the initApp method the user would enter a cycle where the menu would be contained, inside of this cycle the user would choose which action it wanted the program to perform, and the cycle would allow it to continue

interacting the application even after completing an initial task. This class would handle all the main functionalities as well as the user interactions with the program. In order to do so, this class would have to relate to the content classes.

Said content classes consisted of three classes, one abstract class by the name Video, which would be used as a parent class to avoid code duplication and speed up the implementation of features as anything on it would affect its children. This is due to the fact that from reading the necessities of the provider in the problem statement I identified that the platform needed would handle two types of content, which would be movies and series. Because they have similar properties such as a name, an id, etc. I thought I had to create a parent class from which they could both inherit said common attributes and functions. Because of this the Series and Movie classes had an inheritance relation pointing to the Video class because both were constructed to fill the gaps of said abstract class to better represent their real-life counter parts. The principal gap created for them to fill was a `displayInformation()` method, which was a pure virtual function since it was defined as equal to 0 so that the children could fill it as they needed.

The way the `StreamingService` class related to them was that it contained a vector of pointers of Videos, which allowed this class to manipulate said data and populate the vector as the user desired. Because the data was injected into the program through the method `loadData()` and the objects were created and stored inside of this class, this class maintained a composition relation with the Video class, as it stored both Movie and Series objects as Video pointers, which also helped in the way that this class could now call their common method `displayInformation()` and they would still display their specific characteristics. In creating such vector, the inherited operator overloading member function for `==` in Video was used to validate if their rating coincided with what was asked by the user. Both proved very useful when searching for specific ratings and displaying only those who complied with the specific search query provided by the user inside of the menu options. Additionally, the use of the content vector allowed me to store all the data and selectively check if their type coincided with what the user was looking for through the use of the `<typeinfo>` library. Overall, handling the information

this way simplified the relations between classes since I avoided having to create a vector for each type of Video instance the program would contain.

Additionally, because series have episodes I had to program another class that could be used to create instances that correctly represented them and their attributes, because episodes in real life always live inside a bigger context which is a series I decided to create an aggregation relationship between Series and Episode, since Episodes would be contained inside of a vector in series but the series class would not handle their creation I implemented an aggregation relation between them. Since the episode class did not share any resemblance to the other types of content and the provider didn't require them to have functionalities like rating, which is why I assumed the best solution would be to make them their own entity.

Finally, the relation between Episode and StreamingService was of the type use, since episodes didn't live inside said class, they are intended to be stored inside of Series, which is why the StreamingService class uses their constructor to create a vector of episodes that is then passed to the series they belong to as an argument on their constructor when the data is being loaded, which is then pushed into the content vector that lives inside StreamingService instances in the form of pointers.

## Execution example

### 1. Start of the program and menu

```
Cesar@DESKTOP-KJD59H7 MINGW64 ~/Documents/Cplusplus/oop part 2/TC1030 - integrative project - A01703517/src
$ ./a.exe
Welcome to the Streaming Service Content Rating App!!!
--Type anything to enter the streaming service rating App
--Type 1 to exit the program
D

*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: █
```

### 2. Loading data from default file

```

*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 1
You have selected --Load data into App-- . To add your own data, type the path to your data file.
If you want to proceed with the basic data, press 1: 1
-----Data loaded into app successfully-----

```

### 3. Showing currently loaded data.

```

*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 2
Type 1 to only display the content from a certain genre.
Type 2 to display the titles with a certain rating.
Type 3 to search content from a title.
Type 4 to display all content.
4
-----Displaying all available titles-----
****Movie ID:556848456-Title: Terminator, Genre: Action, Rating: 4.5 Studio: Universal, Duration: 120 min.

****Movie ID:452515155-Title: Fast and Furious, Genre: Action, Rating: 2.1 Studio: Universal, Duration: 110 min.

****Movie ID:546465411-Title: Robin Hood, Genre: Adventure, Rating: 4.6 Studio: Disney, Duration: 90. min.

****Movie ID:754524221-Title: Donnie Darko, Genre: Terror, Rating: 3.2 Studio: warner bros, Duration: 125 min.

****Movie ID:855545665-Title: Gremlins, Genre: Terror, Rating: 4.0 Studio: Universal, Duration: 119 min.

****Series ID:989545665-Title: Aliens, Genre: Terror, Rating: 4.0 Seasons: 3, Episodes: 7, Status: currently airing.
Available Episodes:
---Season: 1, Episode: 1, Title: Nostromo
---Season: 1, Episode: 2, Title: The planet
---Season: 2, Episode: 3, Title: The alien
---Season: 2, Episode: 4, Title: The escape
---Season: 3, Episode: 5, Title: Galactic battleground
---Season: 3, Episode: 6, Title: Chestbuster
---Season: 3, Episode: 7, Title: Back on earth

```

```

****Movie ID:451121212-Title: Cinderella, Genre: Fantasy, Rating: 5.0 Studio: Disney, Duration: 90. min.

****Movie ID:451121212-Title: The lord of the rings, Genre: Fantasy, Rating: 5.0 Studio: Universal, Duration: 200 min.

****Movie ID:545812855-Title: The two towers, Genre: Fantasy, Rating: 5.0 Studio: Universal, Duration: 190 min.

****Movie ID:786966958-Title: Return of the king, Genre: Fantasy, Rating: 5.0 Studio: Universal, Duration: 210 min.

****Movie ID:541215847-Title: Beetlejuice, Genre: Comedy, Rating: 2.5 Studio: Universal, Duration: 75. min.

****Movie ID:362514789-Title: Jeepers Creepers, Genre: Terror, Rating: 1.5 Studio: Universal, Duration: 115 min.

****Movie ID:362514789-Title: Sharknado, Genre: Terror, Rating: 0.5 Studio: MGM, Duration: 85. min.

****Movie ID:555452562-Title: Django, Genre: Western, Rating: 3.5 Studio: Universal, Duration: 110 min.

****Movie ID:858512456-Title: Chucky, Genre: Terror, Rating: 0.5 Studio: MGM, Duration: 85. min.

****Movie ID:458566664-Title: The lone Ranger, Genre: Western, Rating: 1.8 Studio: MGM, Duration: 85. min.

****Series ID:989545665-Title: Breaking Bad, Genre: Suspense, Rating: 5.0 Seasons: 2, Episodes: 8, Status: finished.
Available Episodes:
---Season: 1, Episode: 1, Title: Ozymandias
---Season: 1, Episode: 2, Title: Mr white
---Season: 1, Episode: 3, Title: Jesse
---Season: 1, Episode: 4, Title: Los Pollos Hermanos
---Season: 1, Episode: 5, Title: Gray Matter
---Season: 1, Episode: 6, Title: Grilled
---Season: 2, Episode: 7, Title: ABQ
---Season: 2, Episode: 8, Title: One minute

****Series ID:785222154-Title: How I met your mother, Genre: Comedy, Rating: 3.8 Seasons: 1, Episodes: 3, Status: finished.
Available Episodes:
---Season: 1, Episode: 1, Title: The blue instrument
---Season: 1, Episode: 2, Title: Robin and Ted
---Season: 1, Episode: 3, Title: Barney goes to college

```

```

****Series ID:785222154-Title: The good the bad and the ugly, Genre: Western, Rating: 3.8 Seasons: 1, Episodes: 3, Status: currently airing.
Available Episodes:
---Season: 1, Episode: 1, Title: The good
---Season: 1, Episode: 2, Title: The bad
---Season: 1, Episode: 3, Title: The ugly

****Series ID:555556585-Title: Star wars Clone Wars, Genre: Adventure, Rating: 5.0 Seasons: 3, Episodes: 9, Status: finished.
Available Episodes:
---Season: 1, Episode: 1, Title: Rising Malevolence
---Season: 1, Episode: 2, Title: Shadow of Malevolence
---Season: 1, Episode: 3, Title: Destroy Malevolence
---Season: 2, Episode: 4, Title: Rookies
---Season: 2, Episode: 5, Title: Duel of droids
---Season: 2, Episode: 6, Title: Cloak of darkness
---Season: 3, Episode: 7, Title: Lair of Grievous
---Season: 3, Episode: 8, Title: Jedi Crash
---Season: 3, Episode: 9, Title: Storm Over Ryloth

*****Main menu*****

```

#### 4. Loading a new datafile:



For this runtime, the path to the default data file was used, however, this works with files in other directories if the user inputs the global path to the file.

```
*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 1
You have selected --Load data into App-- . To add your own data, type the path to your data file.
If you want to proceed with the basic data, press 1: C:\Users\Cesar\Documents\CPlusPlus\oop part 2\TC1030 - integrative project - A01703517\src\data\content.txt
Data loaded into app successfully

*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 2
Type 1 to only display the content from a certain genre.
Type 2 to display the titles with a certain rating.
Type 3 to search content from a title.
Type 4 to display all content.
4
-----Displaying all available titles-----
***Movie ID:556848456-Title: Terminator, Genre: Action, Rating: 4.5 Studio: Universal, Duration: 120 min.

***Movie ID:452515155-Title: Fast and Furious, Genre: Action, Rating: 2.1 Studio: Universal, Duration: 110 min.
```

## 5. Displaying content by genre

```
*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 2
Type 1 to only display the content from a certain genre.
Type 2 to display the titles with a certain rating.
Type 3 to search content from a title.
Type 4 to display all content.
1
Type the genre you would like to search:
Action
-----Displaying Results-----
***Movie ID:556848456-Title: Terminator, Genre: Action, Rating: 4.5 Studio: Universal, Duration: 120 min.

***Movie ID:452515155-Title: Fast and Furious, Genre: Action, Rating: 2.1 Studio: Universal, Duration: 110 min.

*****Main menu*****
```

## 6. Displaying content by rating

```

*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 2
Type 1 to only display the content from a certain genre.
Type 2 to display the titles with a certain rating.
Type 3 to search content from a title.
Type 4 to display all content.
2
Type the rating you would like to search: (you can only enter 0, 1, 2, 3, 4 or 5)
3
-----Displayin Results-----
****Movie ID:754524221-Title: Donnie Darko, Genre: Terror, Rating: 3.2 Studio: warner bros, Duration: 125 min.

****Movie ID:555452562-Title: Django, Genre: Western, Rating: 3.5 Studio: Universal, Duration: 110 min.

****Series ID:785222154-Title: How I met your mother, Genre: Comedy, Rating: 3.8 Seasons: 1, Episodes: 3, Status: finished.
Available Episodes:
---Season: 1, Episode: 1, Title: The blue instrument
---Season: 1, Episode: 2, Title: Robin and Ted
---Season: 1, Episode: 3, Title: Barney goes to college

****Series ID:785222154-Title: The good the bad and the ugly, Genre: Western, Rating: 3.8 Seasons: 1, Episodes: 3, Status: currently airing.
Available Episodes:
---Season: 1, Episode: 1, Title: The good
---Season: 1, Episode: 2, Title: The bad
---Season: 1, Episode: 3, Title: The ugly

*****Main menu*****

```

## 7. Displaying content by title

```

*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 2
Type 1 to only display the content from a certain genre.
Type 2 to display the titles with a certain rating.
Type 3 to search content from a title.
Type 4 to display all content.
3
Type the title you would like to search:
Aliens
-----Displaying Results-----
****Series ID:989545665-Title: Aliens, Genre: Terror, Rating: 4.0 Seasons: 3, Episodes: 7, Status: currently airing.
Available Episodes:
---Season: 1, Episode: 1, Title: Nostromo
---Season: 1, Episode: 2, Title: The planet
---Season: 2, Episode: 3, Title: The alien
---Season: 2, Episode: 4, Title: The escape
---Season: 3, Episode: 5, Title: Galactic battleground
---Season: 3, Episode: 6, Title: Chestbuster
---Season: 3, Episode: 7, Title: Back on earth

```

## 8. Displaying Series by rating

```

*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 3
Type the rating you would like to search for movies: (you can only enter 0, 1, 2, 3, 4 or 5)
2
-----Displaying Results-----
****Series ID:785222154-Title: The good the bad and the ugly, Genre: Western, Rating: 2.8 Seasons: 1, Episodes: 3, Status: currently airing.
Available Episodes:
---Season: 1, Episode: 1, Title: The good
---Season: 1, Episode: 2, Title: The bad
---Season: 1, Episode: 3, Title: The ugly
*****Main menu*****

```

## 9. Displaying Movies by rating

```

*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 4
Type the rating you would like to search for rating: (you can only enter 0, 1, 2, 3, 4 or 5)
0
-----Displaying Results-----
****Movie ID:362514789-Title: Sharknado, Genre: Terror, Rating: 0.5 Studio: MGM, Duration: 85. min.

****Movie ID:858512456-Title: Chucky, Genre: Terror, Rating: 0.5 Studio: MGM, Duration: 85. min.

*****Main menu*****

```

## 10. Rating content

```

*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 5
Type the title of the content you want to rate:
Title:
Chucky
-----Displaying Results-----
***Movie ID:858512456-Title: Chucky, Genre: Terror, Rating: 0.5 Studio: MGM, Duration: 85. min.

Type the rating you want to give to this content: (you can only enter 0, 1, 2, 3, 4 or 5): 5
Updated content:
-----Displaying Results-----
***Movie ID:858512456-Title: Chucky, Genre: Terror, Rating: 2.7 Studio: MGM, Duration: 85. min.

*****Main menu*****

```

## 11. Exiting app

```

*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 6
Welcome to the Streaming Service Content Rating App!!!
--Type anything to enter the streaming service rating App
--Type 1 to exit the program
1
We appreciate your reviews, come back soon!!!

Cesar@DESKTOP-KJD59H7 MINGW64 ~/Documents/CPlusPlus/oop part 2/TC1030 - integrative project - A01703517/src
$ 

```

## Solution justification

- The proper classes are identified

On the problem statement it is said that the objective is to create a platform that incorporates both movies and series, given that both share the same basic attributes

but the user interacts in a different way with each of them I identified that the best solution would be to create a Video class from which both Movie and Series would derive, this allowed me to easily tinker with what they shared while also modifying the classes that inherited from this class so that they could function properly. Additionally, given the fact that series had episodes that functioned as their own entity I realized that the best solution was to create an Episode class to create the objects that each series needed to represent it's length, which I connected through aggregation. Moreover, I identified that there was a need for a class that held all the application information and had methods that interacted with this information to present it to the user, thus, I think that creating a StreamingService class that held all the functionality was the best option since it allowed me to better manipulate the information flow from one place and handle all user interactions with the data.

- Inheritance is implemented properly

Seeing the interactions and similarities between Series and Movies, I think that the code I implemented was the best solution. This comes from the fact that making their common attributes come from the same place made manipulating both at the same time easier while also allowing me to reuse the code implemented in the video superclass. Additionally, by implementing inheritance I was able to code a method called rate() which they would both inherit since both series and movies had the same rating system. Moreover, the decision to make Video an abstract class came to the fact that it provided the advantage of writing shorter code for child classes by avoiding duplication and allowing me to reuse the functionality of said parent class in the child classes, which is why this was the best implementation of inheritance.

- Access modifiers are implemented properly

When complementing inheritance access modifiers proved to be very important as their correct implementation allowed me to access both the methods and attributes from the parent class inside of subclasses. My codes implementation of this Object Oriented Programming concept was the best because it meant that only getters could retrieve information from each object and that there was no way to change the data other than from an established method like rateVideo() which altered the actualRating

and numberOfVotes attributes. Encapsulating the other attributes was the best decision option better control the information contained inside each object since there was no reason for the user or the program to manipulate attributes such as the title or the release year in the case of a Movie as these were facts. Additionally, when implementing inheritance the solution of making the attributes protected and the methods of the base class public was the best solution because when a child class such as Movie or Series inherited from them publicly I could still use the same methods to interact with the child classes and encapsulation maintained it's purpose even in child classes.

- Method overriding is implemented properly

By reading the problem statement it can be seen that the program is required to allow the same action to be performed on both Series and Videos, such action was displaying their individual information. I think that my solution was the best approach to said challenge because I defined a pure virtual function in the Video class that was overridden with the reserved word override inside of each child class so that each of them could construct their own string with the information that best represented themselves and then return it to be displayed with the help of cout inside of the StreamingService class. Thus, method overriding was implemented properly as the child classes redefined the method and gave it the functionality that constructed the correct display information for them.

- Polymorphism is implemented properly

Apart from implementing method overriding, which is a form of run-time polymorphism, another opportunity area was exploited, such area was that a requisite for the program was to search for specific ratings, genres, and titles from all the catalogue, which meant I had to consider both Series and Movies. I think that my solution was the best I could come up with since I avoided having to iterate through a vector of instances of each different object. I avoided this by having a content vector of Video pointers that referenced the real instances of each Movie and Series. By declaring content I was able to iterate through all the content contained in the StreamingService regardless of their type and call the displayInformation() method

when its information coincided with the search query. By using this I created a dynamic vector of pointers of objects of type Video that reacted according to their own definitions of the overridden method `displayInformation()` since it was declared as a pure virtual function inside of Video. This was accomplished using the `createAddMovies` and `createAddVideos` methods, which created the pointers that were pushed into the content vector. Additionally, by using this solution I avoided having to separate the different subclasses to solve the problem of only displaying movies or series with a certain rating since I was able to use the `typeid` library to check for the actual type of a pointer inside of the content vector, which I used to selectively display only the content of the type selected with the selected rating.

- Abstract classes are implemented properly

After I had constructed my initial classes I decided to make the class Video an abstract class with `displayInformation()` as a pure virtual function that would be defined in a different for each child class so that it represented their individual information better. I think this was the best solution because it was pointless to implement such functionality inside of a class whose only purposes were to serve as a parent to avoid code duplication and have its `displayInformation()` method overridden by its children, by abstracting this class I was able to create a generic mold that defined the basic required functionality that child classes would have to define, which would make creating a Documentary or StandUp class that derived from it easier, since by turning this into an abstract class I clarified what their children would have to override in order to fill in the gaps of the partial description of their parent and accomplish what would be expected from the program.

- At least one operator is overloaded properly

Firstly, I had implemented the functionality of showing the content with a specific rating by creating a complex condition that displayed the information for said content if the rating of a video was between the specified rating inside the query and the rating plus one. For example, the content with a rating between 3 and 4 would be showed if the rating selected was 3. Due to the fact that this condition was complex since the `getActualRating()` method was needed to extract said rating and then compare it to the

rating from the query, I decided and implement what I think was the best solution to simplify this condition. Said solution was to overload the == operator to create an special meaning for it for all Video data types, this was done so that the operation could be handled inside a member function that checked the same condition by receiving the rating as a parameter and comparing it to the actualRating attribute to determine if the current content was inside of the range of ratings that the user wanted to display, in which case this member function returned true. This was the best solution because it allowed me to simplify the conditions used to validate if the rating of a Video complied with said criteria by abstracting the logic behind it and reducing it to the simple use of and == operator between the Video object and an integer that represented the rating.

## Functionality problems

The next functionality problems are validated by the program, however, at the end of this section the only case in which the app could break is identified.

- **Starting App**

Since any miss input in the initialization menu of the App would count as the user trying to start the app, there are no possible errors for the first input.

```
Cesar@DESKTOP-KJD59H7 MINGW64 ~/Documents/Cplusplus/oop part 2/TC1030 - integrative project - A01703517/src
$ ./a.exe
Welcome to the Streaming Service Content Rating App!!!
--Type anything to enter the streaming service rating App
--Type 1 to exit the program
fsfaoadg

*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
```

- **File handling**

Since the same system of validation from the last point was implemented for the selection of an option for the menu option 1, the validation works the same way, it just checks if the extension is .txt



```
*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 1
You have selected --Load data into App-- . To add your own data, type the path to your data file.
If you want to proceed with the basic data, press 1: djdfhakahdd
The file selected was no of type txt, please type the path to a txt file next time.

*****Main menu*****
```

Case 2: the file is in txt format, but its path is not valid.

The path used to test this functionality was: C:\Users\Cesar\Documents\CPlusPlus\oop part 2\TC1030 - integrative project - A01703517\src\data\content.txt

It is important to note that this file does not exist.

```
*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 1
You have selected --Load data into App-- . To add your own data, type the path to your data file.
If you want to proceed with the basic data, press 1: C:\Users\Cesar\Documents\CPlusPlus\oop part 2\TC1030 - integrative project - A01703517\src\data\badpath.txt
ERROR: File could not be loaded, try again with a different path.

*****Main menu*****
```

- Error in main menu input

```
*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: gaga
Your answer was not a valid option, please try again.

*****Main menu*****
```

If the user introduces an option that is not allowed for the menu, then it will only display that said option is not valid and print the menu again since this is contained inside of a while loop that only breaks if the user types 6.

- **Menu option 2**

```
*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 2
Type 1 to only display the content from a certain genre.
Type 2 to display the titles with a certain rating.
Type 3 to search content from a title.
Type 4 to display all content.
gagagaga
Option not available. Returning to menu...

*****Main menu*****
```

The same applies to this submenu, the only difference is that when it detects an invalid input it returns the user to the main menu by printing it again.

- **Search not found for search options.**

The next outputs are also used when data has not been loaded into the app yet. A cycle checks if there is any Video that coincides with the query and prints the information it contains if that is the case, if not, it shows the message that there weren't matches for what the user typed in the terminal.

```

*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 2
Type 1 to only display the content from a certain genre.
Type 2 to display the titles with a certain rating.
Type 3 to search content from a title.
Type 4 to display all content.
1
Type the genre you would like to search:
Crazy
-----Displaying Results-----
----Ooops, we could not find any match for your search.----

*****Main menu*****

```

```

*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 2
Type 1 to only display the content from a certain genre.
Type 2 to display the titles with a certain rating.
Type 3 to search content from a title.
Type 4 to display all content.
2
Type the rating you would like to search: (you can only enter 0, 1, 2, 3, 4 or 5)
65
The rating you selected was not an option. Returning to menu...

*****Main menu*****

```

```

*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 2
Type 1 to only display the content from a certain genre.
Type 2 to display the titles with a certain rating.
Type 3 to search content from a title.
Type 4 to display all content.
3
Type the title you would like to search:
Freddy Kruegger
-----Displaying Results-----
----Oops, we could not find any match for your search.----

*****Main menu*****

```

```

*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 2
Type 1 to only display the content from a certain genre.
Type 2 to display the titles with a certain rating.
Type 3 to search content from a title.
Type 4 to display all content.
jhdgajgsh
Option not available. Returning to menu...

*****Main menu*****

```

- Menu option 3 errors

```

*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 3
Type the rating you would like to search for movies: (you can only enter 0, 1, 2, 3, 4 or 5)
6.4
The rating you selected was not an option. Returning to menu...

*****Main menu*****

```

If the user types anything that is different from the specified options declared in the code the program will not continue with this operation and return to the menu.

- **Menu option 4 errors**

```

*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 4
Type the rating you would like to search for rating: (you can only enter 0, 1, 2, 3, 4 or 5)
test
The rating you selected was not an option. Returning to menu...

*****Main menu*****

```

The same logic from option 3 on the menu applies for inputs that are different from what is stated.

- Menu option 5 errors

```
*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 5
Type the title of the content you want to rate:
Title:
Chucky
-----Displaying Results-----
****Movie ID:858512456-Title: Chucky, Genre: Terror, Rating: 0.5 Studio: MGM, Duration: 85. min.

Type the rating you want to give to this content: (you can only enter 0, 1, 2, 3, 4 or 5): 6.5
You did not followed the instructions. Returning to menu...

*****Main menu*****
```

```
*****Main menu*****
Select the part of the menu you want to enter by typing its number:
*TIP: Try loading the data before using any other part of the app.
1.- Load data into App
2.- Display select content
3.- Display series by rating
4.- Display movies by rating
5.- Rate content
6.- Exit app
Insert option: 5
Type the title of the content you want to rate:
Title:
hKHSAXHAKD JSDH
-----Displaying Results-----
----Ooops, we could not find any match for your search.----
TIP: Try again with a title that is available. Returning to menu...

*****Main menu*****
```

In addition to these input errors, the App would not function if the data inside of the txt file was not in the expected format since the program would not be able to recognize the data of a movie or a series, therefore, there would not be any data loaded into the app when you use a txt with a different data format. However, since this application was

built as a solution for the provider, it is expected that he will use the hypothetical format, which would make this functionality problem a rare scenario in this application.

## **Conclusion**

In conclusion, this solution was the best I could come up with by using the Object Oriented Programming concepts review in class. This is due to the fact that I implemented each concept to simplify the code that would otherwise be very complex if these were not used, because inheritances, method overriding, operator overloading, polymorphism and inheritance allowed me to enhance the structure of my solution by simplifying the relationships between classes. Thus, this solution will not only comply with what was expected from the provider, but it will also provide an optimized solution constructed with expansion in mind, since the implementation of polymorphism through the vectors of pointers and the creation of a mold for content in the way of an abstract class called video will allow the implementation of new types of content like documentaries. In the end, the process of coding such a complex solution taught me that I should never marry to an idea since the implementation I had in mind at first was not even close to what was programmed in the end. With this project I learned the importance of organizing the structure of a program through a class diagram, since the diagram I created really came to reality thanks to the time and thinking that went into its construction and the relations it would represent, however, it had some changes that simplified it at the end, which is why it is always important to constantly think of new ways of improving one's code as a programmer to make it less confusing and optimized.

## References:

[duplicate], g., Jamshaid, M., & Hammen, D. (2012). getline() does not work if used after some inputs. Retrieved 31 May 2021, from:

<https://stackoverflow.com/questions/12691316/getline-does-not-work-if-used-after-some-inputs>

Applications of Pointers in C/C++ - GeeksforGeeks. (2018). Retrieved 3 June 2021, from <https://www.geeksforgeeks.org/applications-of-pointers-in-c-cpp/>

C++ Files. (2021). Retrieved 31 May 2021, from:

[https://www.w3schools.com/cpp/cpp\\_files.asp](https://www.w3schools.com/cpp/cpp_files.asp)

Input/output with files - C++ Tutorials. (2021). Retrieved 31 May 2021, from

<https://www.cplusplus.com/doc/tutorial/files/>

string::substr - C++ Reference. (2021). Retrieved 31 May 2021, from

<https://www.cplusplus.com/reference/string/string/substr/>

stof - C++ Reference. (2021). Retrieved 31 May 2021, from

<https://www.cplusplus.com/reference/string/stof/>

Operator Overloading in C++ - GeeksforGeeks. (2014). Retrieved 3 June 2021, from

<https://www.geeksforgeeks.org/operator-overloading-c/>

Object Oriented Programming in C++ - GeeksforGeeks. (2015). Retrieved 3 June 2021,

from <https://www.geeksforgeeks.org/object-oriented-programming-in-cpp/>

Inheritance in C++ - GeeksforGeeks. (2017). Retrieved 3 June 2021, from

<https://www.geeksforgeeks.org/inheritance-in-c/>